Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

Graded Quiz: SQL CASE Statements

Latest Submission Grade 100%

1. How many managers are in the employees database? **1 / 1 point**

● 24

○ 26

○ 22

○ 20

✓ **Correct**
The dept_managers table has 24 rows. This means that there are 24 managers in this database.

2. Write a query that retrieves the **player_name, year,** and creates a column called **senior_student** that returns **yes** if the year is **'SR'** and **no** otherwise from the **college_football_players** table. **1 / 1 point**

●

```
SELECT player_name, year,
CASE
    WHEN year = 'SR' THEN 'yes'
    ELSE 'no'
END AS senior_student
FROM college_football_players;
```

○

```
SELECT player_name, year,
CASE
    WHEN year = 'SR' THEN 'yes'
    ELSE 'no'
FROM college_football_players;
```

○

```
SELECT player_name, year,
CASE
    WHEN year = 'SR' THEN 'yes'
    ELSE 'no'
FROM college_football_players
END AS senior_student;
```

○

```
SELECT player_name, year
CASE
    WHEN year = 'SR' THEN 'yes'
    ELSE 'no'
END AS senior_student
FROM college_football_players;
```

✓ **Correct**
Correct! This query retrieves the **player_name, year,** and creates a column called **senior_student** that returns **yes** if the year is **'SR'** and **no** otherwise from the **college_football_players** table.

3. What is the function of the ELSE statement in the CASE statement? **1 / 1 point**

○ It captures values specified in the WHEN/THEN statements

○ It gets the last condition in the END statement

● It captures values not specified in the WHEN/THEN statements

○ It captures values not specified in the IF/THEN statements

✓ **Correct**
Correct! The ELSE statement is optional and provides a
way to capture values not specified in the WHEN / THEN statements.

4. Aggregate functions do not consider null values. **1 / 1 point**

● True

○ False

○ Maybe

✓ **Correct**
Correct! All aggregate functions ignore null values.

5. Which of this type of join returns an output of only matching records in the two tables? ***(Select all that apply)*** **1 / 1 point**

☐ LEFT JOIN

☑ INNER JOIN

✓ **Correct**
Correct! This type of join returns those records which have matching values in both tables.

☑ JOIN

✓ **Correct**
Correct! This type of join returns those records which have matching values in both tables.

☐ RIGHT JOIN

6. Write a query to retrieve a list of the average salary of employees. If the average salary is more than 80000, return Paid Well. If the average salary is less than 80000, return Underpaid, otherwise, return Unpaid **1 / 1 point**

●

```
SELECT emp_no, ROUND(AVG(salary), 3) AS average_salary,
CASE
    WHEN AVG(salary) > 80000 THEN 'Paid Well'
    WHEN AVG(salary) < 80000 THEN 'Underpaid'
ELSE 'Unpaid'
END
FROM salaries
GROUP BY emp_no
ORDER BY average_salary DESC;
```

○

```
SELECT emp_no, ROUND(AVG(salary), 3) AS average_salary,
CASE
    WHEN AVG(salary) > 80000 THEN 'Paid Well'
    WHEN AVG(salary) < 80000 THEN 'Underpaid'
ELSE 'Unpaid'
FROM salaries
END
GROUP BY emp_no
ORDER BY average_salary DESC;
```

○

```
SELECT emp_no, ROUND(AVG(salary), 3) AS average_salary,
CASE
    WHEN AVG(salary) > 80000 THEN 'Paid Well'
    WHEN AVG(salary) < 80000 THEN 'Underpaid'
ELSE 'Unpaid'
END
FROM salaries
ORDER BY average_salary DESC
GROUP BY emp_no;
```

✓ **Correct**
Correct! This query retrieves a list of the average salary of employees. If the average salary is more than 80000, return Paid Well. If the average salary is less than 80000, return Underpaid, otherwise, return Unpaid

7. Which SQL statement is used to sort the result set of a query? **1 / 1 point**

● ORDER BY

○ None of the above

○ HAVING

○ GROUP BY

✓ **Correct**
Correct! The **ORDER BY** statement allows us to sort our results using the data in any column.

8. What will be the result of this query? **1 / 1 point**

```
SELECT a.profit_category, COUNT(*)
FROM (
    SELECT order_line, profit,
    CASE
        WHEN profit < 0 THEN 'No Profit'
        WHEN profit > 0 AND profit < 500 THEN 'Low Profit'
        WHEN profit > 500 AND profit < 1500 THEN 'Good Profit'
        ELSE 'High Profit'
    END AS profit_category
    FROM sales
) a
GROUP BY a.profit_category;
```

○ Retrieves a list of all profit categories from the salaries table

● Retrieves the count of the different profit categories in the sales table

○ Retrieves a list of all profit categories from the customers table

○ Retrieves a list of all profit categories from the sales table

✓ **Correct**
Correct! The query returns the count of the different profit categories in the sales table

9. What is the golden rule for performing SQL joins? ***(Select all that apply)*** **1 / 1 point**

☑ Find the linking field or column in both tables

✓ **Correct**
Correct! One important thing to note is, we can only join these two tables on their related or linking column or field,

☑ Find a related column or field in both tables

✓ **Correct**
Correct! One important thing to note is, we can only join these two tables on their related or linking column or field,

□ Find a related row or record in both tables

□ Find the unique and primary keys in both tables

10. Aggregate functions in SQL are ___________ **1 / 1 point**

● Built-in functions

○ User-defined functions

✓ **Correct**
Correct! The aggregate functions are built-in SQL functions that are used to retrieve summaries of data from database objects.

11. The LIMIT statement is not always the last part of a query. **1 / 1 point**

● False

○ Maybe

○ True

✓ **Correct**
Correct! In SQL, the LIMIT statement is always the last part of a query. It helps to retrieve rows or records of a table as specified.